

Lyee Internet Information

イタリア ピサ大学

ヴィンセント・アンブリオラ教授インタビュー



日時 : 4月20日(金)
場所 : ソフトウェア生産技術研究所株式会社 本社

【ヴィンセント・アンブリオラ(Vincenzo Ambriola)教授 略歴】

1954年生まれ。 現在47歳。
1982年 ピサ大学コンピュータ・サイエンス学部で助教授に就任。
1984年 アメリカ カーネギーメロン大学の客員科学者として
ガンダルフ・プロジェクトに携わる。
1989年 ウディネ大学コンピュータ・サイエンス学部の準教授に就任。
1992年 ピサ大学コンピュータ・サイエンス学部の準教授に就任。
現在 同大学同学部の副学部長及び教授。
Software Process Improvement and Practice の編集委員会
メンバー。
ソフトウェア処理技術に関する欧州ワークショップのプログラム
議長。

【 Lyee の印象 】

- まずは経歴について簡単に教えてください。

アンブリオラ教授

私は現在、ピサ大学コンピュータ・サイエンス学部の副学部長を務めております。またピサ大学の情報システムの責任者でもあり、コンピュータ・サイエンス学部の教授でもあります。今年の11月で副学長職は終了し、教授職だけとなります。

- それでは Lyee を知った経緯について教えてください。

アンブリオラ教授

ハミド先生からピサ大学でプレゼンテーションを行わせて頂きたいというメールを頂いたのがきっかけです。ハミド先生がどうして私の名前やメールアドレスを知ったのかはわかりませんが、その要望を受け入れて、2月にお越し頂いてプレゼンテーションをしてもらいました。この様なリクエストを頂く事はよくあるのですが、そういった場合、大学の学者や学生を集めてプレゼンテーションをして頂いています。2月のプレゼンテーションは日本から同じ分野を研究している、いわば同僚に来て頂いたといった形でした。

それまでは Lyee について全く知らなかったのですが、ソフトウェア生産技術研究所の方にプレゼンテーションをして頂いて初めてこういった方法論の存在を知りました。

- プレゼンテーションを聞いた際に興味を持つことはできましたか？

アンブリオラ教授

プレゼンテーションが行われる前にインターネットを使い情報を探したのですが、十分な情報や資料がなかったため、プレゼンテーションが行われるまで全く未知の状態でした。

私の研究分野はリクワイヤメント・エンジニアリングなのですが、プレゼンテーションを聞いて、自分の研究分野に近いと感じて興味を持ちました。

特に興味を持ったのは、単語を起点とするといった考え方とツールからコードを自動生成するという点です。

私が行っている研究もよく似たアプローチをとっておりまして、まず顧客にシステムを顧客自身の言葉を使って定義してもらって、その後我々がコードを自動生成するのです。

また私はソフトウェア・プロセス・モデリングの分野も研究しておりまして、Lyee がどのようなプロセスでソフトウェアを開発しているのかについて知りたいと思いました。また Lyee が

どのようなコンピューショナル・モデルを使っているのかについて興味を持ちました。

- 先生が行っている研究も Lyee によく似たアプローチをとられているとおっしゃれましたが、大きな違いはどこにあるのでしょうか？

アンブリオラ教授

私の行っている研究の目標は、システムの一般的な構造を定義する事で、下位のレベルには焦点をあてておりません。

例えば部屋の中で動くロボットをコントロールするシステムの定義をする場合、ロボットを動かすために外からエンジンに送られてくるシグナルやロボットの動作といった点からシステムの定義を行います。

我々のアプローチでは、ロボットの次の動きを厳密に計算するためにシステムが使うアルゴリズムは記述しません。

我々はコンポーネントという観点から上位の定義を行っている訳で、コードという観点における下位の部分は扱っておりません。

一方 Lyee は外界とアプリケーション間の詳細なインタラクションを定義しています。そういう意味では2つのアプローチは違う問題を取り扱っていますが、相補的とも言えます。

- Lyee もユーザーの意図を客体化するという意味では、詳細部分だけではないと言えるのではないのでしょうか？

アンブリオラ教授

根来さん（ソフトウェア生産技術研究所 社長）の意図は十分理解しておりますが、私は根来さんのアプローチというのがユニバーサルではないという証拠を示そうとしました。

Lyee で作る事が可能なアプリケーションはたくさんあるのでしようけれども、今のところビジネス・アプリケーションが多く、それで充分だと思います。

- 今回来日されて初めて詳細な説明を聞いて、第1印象と違うと感じたところはございますでしょうか？

アンブリオラ教授

来日して最初に聞いた根来さんの説明は、ピサ大学でのプレゼンテーションの内容を確認する様なものでしたが、今日この方法の価値を見たような気がします。

今朝多摩ソフトウェア工場を見学に行き、この考え方が実際のプロセスに変換され、それを使って開発を行っているという事

を自分の目で見ることができました。ソフトウェア工場については非常に興味をもちました。と言いますのも、今現在イタリアで100名のスタッフがいるソフトウェア工場では私はコンサルティングをしているからです。

- イタリアで設立されたソフトウェア工場も Lyee のソフトウェア工場と同じ様な形式なのでしょうか？

アンブリオラ教授 大きな銀行のソフトハウスが設立したもので、銀行のアプリケーションのコードをそこで生成しています。

【 Lyee のアプローチに対する評価 】

- Lyee のアプローチで特に気に入った部分はどこでしょうか？

アンブリオラ教授 Lyee はユーザー・インターフェースからアプリケーションを作り出す、ある1つの方法だと思います。このカテゴリーに属するアプローチは、他にも数多くあります。
私が理解している Lyee のポイントは、全く異なるスキルを持った人達が、別の問題解決の為に作業が出来るという事です。
例えば数名の人が分析作業だけに携わって、他の多くの人がシステムの記述を行うテーブルの作成などができると思います。
我々がイタリアのソフトウェア工場で使っているアプローチは高いスキルの人がたくさん必要なもので、高スキルの人を集めるだけでも大変です。
そういった点において Lyee は他の方法に比べて、リソースを効率的に使う事ができると思います。

- Lyee は開発を行う上で、処理順序を考慮しなくても開発を行う事ができるのですが、このアプローチについてはどの様に評価しますか？

アンブリオラ教授 これはコードを生成していく方法の1つで、他にも同じ様なアプローチはありますので、あまり驚きませんでした。例えば宣言型 (declarative) といったアプローチがありまして、これも順序性を考慮しなくても開発を行う事ができます。

- 宣言型も Lyee の様にイタレーションによって順序性をなくしているのでしょうか？

アンブリオラ教授

Prolog も宣言型アプローチの 1 つです。ML もそうです。

プログラム言語には宣言型というカテゴリーに分類できるものがたくさんあります。

- Lyee は Prolog に似ているとおっしゃられる人もありますが、その意見についてはどう思いますか？

アンブリオラ教授

その意見は間違っていると思います。

Prolog の基本はロジック・プログラミングなのですが、根来さんの定義によると、Lyee はロジックに基づいていないので、両者は違うものだと思います。

もちろん私のいる大学では世界中の Prolog に関する専門家がおり、私も Prolog を使ったプロジェクトを行いましたので、Prolog について聞いただけの知識で言っているのではありません。

- データ項目毎に独立したプログラムを作成していくというアプローチについては、どう評価していますか？

アンブリオラ教授

Lyee ではプログラムは、画面や帳票から始まると聞きました。つまり各画面・帳票に対して出力・入力の関係に基づいてビヘイビアを定義しなければなりません。従って全体のシステムはこの定義を使って作られます。

質問されたのは 1 つの画面を修正する際に、他の画面に手を加えなくてもよいという事だと思いますが、それはシステムが独立したユニットで作られているという事になります。そういった意味では、おっしゃった事は正しいと思います。

こういった特性はモジュラリティーと呼ばれ、他の言語にもこの特性はあります。

オブジェクト指向でも同じで、オブジェクトに対してモジュラリティーというものがあります。Lyee はソフトウェアの世界に既にある考え方を採用しているのだと思います。

- Lyee とオブジェクト指向を混同される方がいるのですが、その意見についてはいかがでしょうか？

アンブリオラ教授

簡単に違いを説明するのは難しいのですが、例えばシステム・デザインをする際に、オブジェクト指向ではアプリケーション・ドメインにある全てのエンティティを定義します。そしてそのエンティティをシステムの中でオブジェクトとして表現していきます。

今朝多摩ソフトウェア工場で見たのですが、Lyee はデータベース・スキーマの点からアプリケーション・ドメインのオブジェクトをモデル化しています。

そして、そこに出力や入力などのビヘイビアを定義しています。ですので、この2つは違うものだと思います。

もちろんこれはどういったアプリケーションを開発したいかによって、どちらを使うかを選択すればよいのです。

- シナリオ関数のメカニズムについてはどう評価しますか？

アンブリオラ教授

その部分に関してはあまり詳細な説明がありませんでしたので、今は自動的にコードを生成する為のメカニズムなのだろうという事しか言えません。

説明が明確でなく充分でもなかったのが、科学的な評価を今してくれと言われても、出来ないのが現状です。しかし実際にこの方法を使って、システムが動いていますので大丈夫なのだと思います。

更に根来さんがお使いになっている用語が、他の方法論で使われている用語と全然違うので、用語が難しいというのではなく、プレゼンテーションを理解するのが困難だと思います。

我々のコミュニティで使われている標準的な方法をお使いになることをお勧めします。

また私はプログラミング・セマンティックスというものを行っておりまして、そこではプログラミング言語の意味を表現するのに4つの異なる方法で表現できます。これは長い歴史があり、非常にアクティブで、現在存在する主要な言語を表現する事ができます。この中には Java も含まれています。

根来さんが何故他の用語を使って表現しないのか、独自の用語に固執しているのかが理解出来ません。これでは先のない道になってしまいます。

- 従来法ではプログラムの共通化、部品化を行います。Lyee では全く反対で共通化を行いませんが、このアプローチについてはどう思われますか？

アンブリオラ教授 根来さんは Lyee におけるデザイン・アクティビティの存在を否定されていますが、Lyee は T0 や T1 といったチャートを作り、そして画面や帳票などといった他の部分を作り、それらを全てまとめた形で構成していく訳で、システムのデザインを行っています。これはシステム・デザインを上流レベルで行っているという事です。システムのデザインを行うときに各要素を明確にし、各モジュールに対して入出力機能を表す各要素を定義します。ですから、これはモジュール定義とアーキテクチャ・デザインです。

- 共通化を行わないので影響調査を行わなくてもよくなり、メンテナンス効率がよくなるというのが Lyee の特徴なのですが、その点についてはいかがですか？

アンブリオラ教授 画面のある項目に出力されるデータを変更するといった事は、LyeeALL を使って定義を変えるだけでよいので、Lyee では簡単にできると思います。但し、システムの内部でミスがあった場合、全てを変更しなければなりません。Lyee がこの部分の修正作業を簡単にしてくれる訳ではないと思います。画面の出力などの部分に焦点を置きますと確かにメンテナンスは非常に簡単になると思いますが、全体を見た時にはそうはならないと思います。

- 先生が全体とおっしゃられているのは、アプリケーション全体という意味ではなく、OS やハードウェア環境などを含めて全体とおっしゃられているのでしょうか？

アンブリオラ教授 私がお話ししているのは、異なるアイテムから作られているアプリケーション全体のアーキテクチャの事です。ここでは異なるコンポーネントがチャンネルで繋がっています。Lyee ではこれらのチャンネルを定義しません。例えば Web ページをデザインする Web とツールがあるとします。そして Web ページの生成に問題がある場合、ツールを使って何かを変更します。データベース・スキーマに問題がある場合は、スキーマの中に何かを修正しなければなりません。もしアーキテクチャそのものにミスがある場合、もしくはアーキテクチャを変更したい場合には、全てを変えなくてははいけません。私は根来さんにこのアーキテクチャを Lyee で作って頂きたい

です。根来さんのアプローチによれば、単語を使ってアーキテクチャが作れるはずですが、それはできないと思います。

Lyee という方法論の範囲はユーザー・インターフェース、データベース・アクセスのためのアプリケーションの生成だと思えます。

Eコマース・アプリケーションでは、プロトコルや暗号化が必要になってきますが、Lyee ではその様な問題は取り扱っていないと思います。

- Lyee は各データ項目に値が入るまで処理を繰り返しますが、この点について、銀行などの大規模システムなどではレスポンスの問題を懸念される方がいるのですが、その点についてはいかがでしょうか？

アンブリオラ教授

業務アプリケーションで使われているロジックというものは、データ等を動かすといったロジックがメインだと思います。値の代入や計算といったシンプルなものですので、イタレーションしてもそんなに時間はかからないので、問題はないと思います。

しかし別の問題があると思います。LyeeALL には、計算式のコードを記述するボックスがあるのを見ました。例えば COBOL を使って記述することもできます。もしここでループを記述するとシステムはそれを受け入れます。そしてループが終了しなければプログラムも終了しません。これらのコネクシオンの安定度を計算していないことが問題なのです。

つまり危険な終了しない COBOL コードを挿入してしまうリスクを負うことになります。複雑なアルゴリズムを計算する関数かサブルーチンがあるとします。それをボックスに入れますが、このサブルーチンがうまくできているという証拠は外側にあるのです。これは Lyee の問題ではなく、サブルーチンを書いている人の問題です。

- Lyee ではデータベース設計を行わずに開発を行い、必要であれば後でまとめるという方法をとりますが、その点についてはどう思われますか？

アンブリオラ教授

そのような説明の仕方は正確ではないと思います。例えばここではオラクル・データベースを使っているのに、他のメソッドと変わりはないと思います。インクリメンタルな方法をとっていると思うのですけれども、ファイルを定義することに関して Lyee

では特別なアクティビティを行います。
つまり別のやり方で同じ事を行っているのだと思います。

以上

* 当内容の無断転載を禁じます。

* Copyright (c)2001 CATENA CORPORATION